

열차 발착시간에 대한 열차 운용 스케줄링 시스템

유영훈* · 황종규** · 조근식***

Train Operations Scheduling System for Train Time-Table

Young-Hoon Yu* · Jong-Gyu Hwang** · Geun-Sik Jo***

요 약

열차운용이라 함은 열차의 출발과 도착 시간을 나타내는 열차번호들에 여러 객차들로 구성된 하나의 열차를 일컫는 열차 다이어아(train DIAGram)들을 여러 제약조건을 고려하여 배정함으로써 열차 번호가 열차 다이어아에 의해 운행될 수 있도록 열차운용 다이어아를 작성하는 작업이다. 이러한 작업을 다수의 스케줄링 전문가들이 스케줄링 할 경우 많은 시간과 노력을 감수해야만 하며, 전문가의 실수로 인해 오류가 발생할 가능성이 있다. 뿐만 아니라, 복잡한 스케줄의 특성상 운용 비용이 적게 드는 스케줄 결과를 이끌어 내기가 어렵다. 본 연구에서는 열차운용 시스템을 구성하기 위해서 현재 철도청에서 사용하는 시스템을 분석하여, 열차운용 스케줄링 시스템을 모델링 하였고, 이에 모델링한 시스템을 바탕으로 CSP(Constraint Satisfaction Problems: 제약조건 만족 문제) 해결 기법을 적용하여 계산시간 및 계산 복잡성 면에서 효율적인 스케줄링을 행함과 동시에 비용 최적 함수를 적용함으로써 열차 운용 비용이 최소가 되는 열차운용 스케줄링 시스템을 구현하였다. 더불어, 시스템 모델 검증과정의 하나로 과거 운용되었던 새마을호 열차시간표와 열차 데이터를 사용하여 열차 운용 및 운용 비용의 최적화를 실험하였다.

Key Words: 열차운용, 스케줄링, CSP, 전문가 시스템

* 인하대학교 전자계산공학과

** 한국철도기술연구원 철도신호통신팀

*** 인하대학교 전자계산공학과

1. 서 론

열차운용 스케줄링 문제는 열차의 출발시간과 도착시간, 출발장소와 도착장소들로 구성된 시간표(Time Table)인 열차번호들에 운행하는 열차를 일컫는 열차 다이어(Train Diagram)들을 배정하는 문제이며, 이때 열차운용 스케줄링 전문가가 사용하는 제약조건들을 추출하여 스케줄링시 고려되어야 하는 문제이다. 이러한 열차 배정 문제는 전형적인 스케줄링 문제로서 n 개의 열차번호들에 m 개의 열차 DIA들을 스케줄링 할 경우 $O(n! \times m)$ 의 매우 큰 복잡도를 갖는 NP 문제이다. 게다가, 이 문제에서는 열차 운용의 특성상 시간표에 의해 열차가 왕복 운행될 수도 있으며, 시간 축에 의해 열차를 배정해야하는 시간 추론 메커니즘의 특성도 가지고 있다. 또한, 적은 수의 열차로 많은 시간표를 운용하기 위해서는 각 역에 도착한 열차들을 시간 축에 의해서 스케줄 하도록 고려되어야 한다. 이와 같은 어려움으로 인해 실제로, 인간 전문가가 스케줄하기 위해서는 많은 노력과 시간이 소요되고 있으며, 많은 사전 도메인 지식이 요구되고 있다.

전통적으로 연구자들은 이런 종류의 문제를 풀기 위해 수리계획법을 사용해왔다. 그러나, 수학적 변수만으로 이루어진 도메인 지식과 제약조건을 모델링(constraints modeling)으로는 문제를 풀기가 매우 어렵다. 최근에는 많은 연구자들이 이러한 문제를 푸는데 CSP(Constraint Satisfaction Problems : 제약조건 만족 문제)와 같은 인공지능 기법의 사용을 제안했다(Tsang, 1993; Pascal, 1989). 인공지능 기법은 복잡한 스케줄링 문제를 모델링 하는데 있어서 수리계획법보다 더 나은 융통성과 효율성을 제공한다(양중윤, 1998.2; Dhar et al. 1990).

본 논문에서는 현재 철도청에서 사용되는 열

차 운용 시스템을 분석 후, 필요한 도메인과 제약조건을 추출하였다. 여기에 CSP를 적용하여 문제를 모델링하였으며, 열차 운용 스케줄링 시스템을 구현하였다.

2. CSP 스케줄링 문제

스케줄링 문제는 각각의 문제가 갖는 특성과 목적에 따라 차이가 발생할 수 있으나, 일반적으로 스케줄링은 어떤 제약조건을 만족하도록 m 개의 자원을 n 개의 작업에 할당하는 문제로서 계산 복잡도상에서 보면 NP-hard류에 속한다고 볼 수 있다. 또한 m 개의 자원을 n 개의 작업에 할당할 경우, 가능한 스케줄링 방법은 $(m!)$ 가지 수가 존재하게된다. 따라서 문제의 규모가 커질수록 작업과 자원 사이에는 복잡한 여러 가지 다른 형태의 제약 조건이 상호 연결되어있기 때문에 해결하기 위한 효과적인 방법이 요구되며, 해결방법 또한 더욱 어려워진다.

전통적으로 스케줄링 문제에 대한 연구는 OR (Operations Research)분야에서 꾸준히 연구되어 왔다. 그중 스케줄링 문제를 해결하는데 있어서 선형계획법(LP, Linear Programming)이 주로 사용되었다. LP의 기호적 제약조건(Symbolic Constraint)과 수치적 제약조건(Numerical Constraint)으로 이루어진 스케줄링 문제는 정수 프로그래밍(Integer Programming) 방법으로 모델링 될 수 있다(Dhar et al. 1990). 본 논문과 유사한 문제가 정수형 프로그래밍 기법(Integer Programming)으로 연구되기도 하였는데, 네덜란드에서는 철도 스케줄을 모델로 하여, 열차 승무원 스케줄링 문제와 차량할당 스케줄링 문제와 같은 과제들이 연구되기도 하였다(Schrijver, 1993; Ernesto et al, 1997; Murphy et al, 1997).

그러나, 정수 프로그래밍은 스케줄링 문제의

규모가 커지면 커질수록 방대한 양의 변수들이 필요할 뿐만 아니라, 이들 사이의 복잡한 상호 관계가 요구되고, 모델링 하는 것도 복잡하고 어려운 과정을 거쳐야 한다. 특히 정수 프로그래밍 방법으로 모델링 할 경우 제약 조건의 논리곱(conjunction) 관계는 표현하기가 용이하나, 논리합(disjunction) 관계의 경우에는 표현과 정형화(formulation)가 어렵고, 기호적 제약 조건을 정수 프로그래밍으로 해결하기 위해서는 많은 양의 인위 변수(artificial variables)가 필요하게 된다. 뿐만 아니라, 이러한 복잡한 모델링 작업은 시스템에 대한 수정을 어렵게 하는 한계를 가지고 있다(Dhar *et al.* 1990).

위와 같이 OR기법은 주로 수치적 제약 조건에 의존하여 스케줄링 문제를 해결하는 반면에 AI 기법은 주로 기호적 제약조건을 다루기에 효율적이다. AI의 스케줄링 문제에 대한 여러 가지 해결방법 중 스케줄링 문제를 CSP(Constraint Satisfaction Problem : 제약조건 만족 문제)로 정의하여 해결하는 방법에 대한 연구가 활발히 전개되고 있다.

CSP는 유한 이산 도메인을 가지는 변수들의 집합과 이들 변수들 사이의 제약조건들의 집합으로 문제를 정형화한 후, 불필요한 도메인을 여과하여 탐색공간을 축소시킴으로써 탐색시간을 축소시키는 제약조건 일치성 검사(Constraint Consistency Checking)와 휴리스틱 검색 알고리즘(Heuristic Search Algorithm)등을 적용하여 문제를 해결하는 방법이다(Tsang, 1993; Pascal, 1989; Wallace, 1994; Mackworth, 1997, Frost, 1994; Nadel, 1989). 이 방법은 주로 스케줄링, 자원배정, 회로설계, 그래프문제 등에 응용된다.

이와 같이, 문제들을 수학적 프로그래밍이 아닌 CSP로서 나타내고 푸는 이유는 첫째, CSP의 변수들은 문제에서의 개체(entity)에 직접 대응되

며 제약조건이 수학적 부등호로 나타내어질 필요 없이 바로 표현 가능하기 때문이다. 이는 형식적인 면에서 간단해지며 이해가 쉬울 뿐만 아니라 좋은 휴리스틱의 선택도 직관적으로 이루어질 수 있다. 두 번째는 비록 단순한 수준의 CSP 알고리즘을 사용한다 하더라도 때로는 정수형 프로그래밍 방법보다 더욱 빨리 해를 찾기 때문이다(양중윤, 1998; Dhar *et al.*, 1990).

여러 분야에서, 이러한 AI 기법으로 스케줄링 문제를 다룬 많은 전문가 시스템이 소개되고 있다. RACES라는 시스템은 많은 항공기들을 복잡한 제약조건하에 주기장에 주기 시키는 시스템으로 현업에 적용되어 사용되고 있다(Jo *et al.* 1997). 또한, 열차 스케줄링 문제에 있어서 열차역과 철도 분기 선로를 관리하는 분산 시스템을 ECRC Prolog의 확장 도구인 SEPIA 논리 프로그래밍 언어로 구현한 연구가 수행되기도 하였다(Lamma *et al.*, 1995).

최근에는 제약조건 프로그래밍 도구의 개발로 문제를 CSP로 표현하고 해결하는데 더욱 용이하게 되었다. 대표적인 도구로는 Prolog를 확장한 제약 논리 프로그래밍 언어인 CHIP(Constraint Handling In Prolog)이 있으며, C++ 라이브러리 형태인 ILOG사의 ILOG Solver와 ILOG Scheduler와 같은 개발 도구들이 있다.

본 연구에서는 ILOG사의 ILOG Scheduler와 ILOG Solver 및 ILOG View를 개발도구로 사용하였다.

3. 열차운용 스케줄링 시스템

3.1. 철도 시스템 분석

현재 한국 철도청의 열차를 운용하기 위한 시

시스템은 다음과 같은 흐름을 가지고 있다.

각 역으로부터의 수송 수요와 기본 수송계획으로부터 열차 계획을 수립하고, 열차 시간표를 구성한다. 만들어진 열차 시간표와 각 차량 사무소의 기본 열차정보를 바탕으로 가상 열차가 운용될 열차 시간표들을 스케줄링 하여 열차 운용 DIA를 만들게 된다. 여기서 가상 열차(Virtual Train)는 실제 객차와 동력차로 구성된 열차가 아닌, 실제 객차와 동력차가 배정될 임의의 가상 열차이다. 이 가상 열차를 열차 DIA(Train DIAGram)라고 부르며, 각각 고유한 번호들이 부여된다. 이러한 열차 DIA들로 구성된 테이블을 열차 운용 DIA 표라고 하며, 이는 각 열차들의 출발역과 도착역을 나타내는 운행 행로와 출발 시간과 도착시간을 나타내는 운행 시간표들로 구성된다. 이러한 열차 운용 DIA 표를 바탕으로 실제로 열차가 운행되기 위한 객차와 동력차를 배정하는 차량 할당 스케줄링을 한다. 또한, 열차 운용 DIA 표와 각 승무원 사무소의 승무원 정보를 바탕으로 각 승무원들이 언제 어디서 어

떤 열차를 운행해야 할지를 결정하는 승무원 운용 스케줄링을 하게 된다.

앞에서와 같은 철도 시스템에서, 결국, 열차운용 시스템은 열차 시간표에 가상 열차 DIA들을 여러 제약조건을 고려하여 스케줄링 하여 열차 운용 DIA 표를 구성하는 문제라고 할 수 있으며, 이는 차량 운용 스케줄링 시스템과 승무원 운용 스케줄링 시스템의 기본 데이터가 된다. [그림 1]은 이러한 철도 시스템의 전체 개요도를 보여주고 있다.

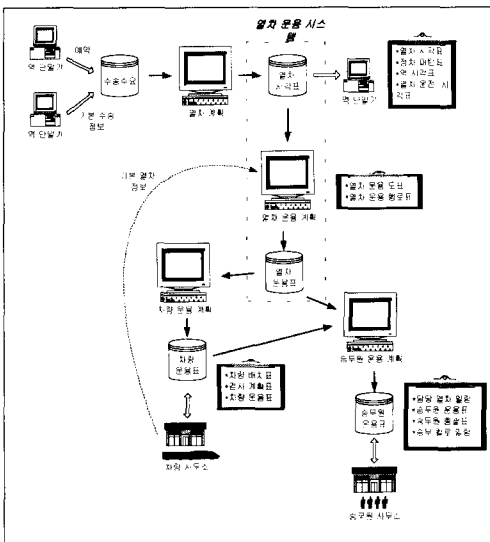
3.2 열차 운용 시스템 분석

열차 운용 시스템은 아래와 같은 데이터를 기본 데이터로 하고, 기존 전문가의 지식을 적용하여 열차 운용 DIA 도표를 출력하는 문제를 다루는 시스템이다.

- 열차번호(Train Number) : 열차가 몇 시에 어디에서 출발하여 몇 시에 어디에 도착하는 운행을 나타내는 열차 시간표들 각각에 부여된 번호.
- 열차 DIA 번호 : 운행할 동력차 및 차량들로 구성되었다고 가정한 가상의 열차들에 부여된 번호.
- 각 출발역에 있는 열차 정보 : 각 출발역에 있는 운행 가능한 열차 수.

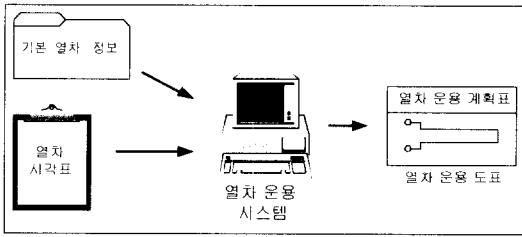
[그림 2]는 이러한 열차 운용 시스템의 개요도를 보여주고 있다.

여기서, 열차 DIA에 실제 객차와 동력차를 배정하는 문제는 차량 스케줄링 문제로 본 연구에서는 고려하지 않았다. 결국, 열차 운용은 열차호에 열차 DIA를 배정하는 것을 문제로 한다.



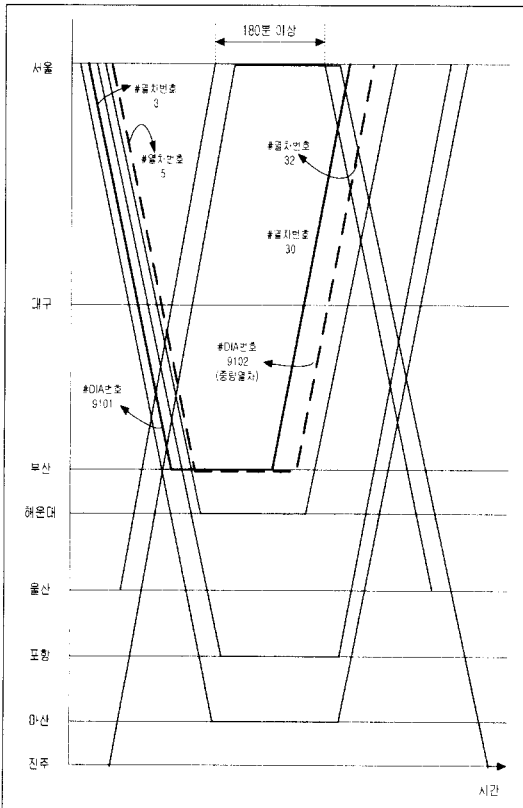
(그림 1) 철도 시스템의 전체 개요도

이러한 DIA번호들은 2개 이상의 열차번호들로 구성되어 있다.



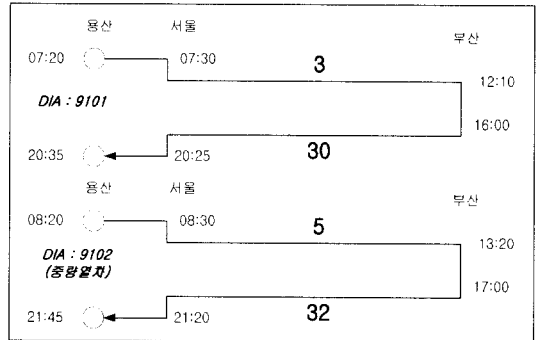
(그림 2) 열차 운용 시스템 개요도

예를 들어, 다음 [그림 3]은 복잡한 열차 DIA도표 중에서 일부만을 나타낸 것이다.



(그림 3) 열차 DIA 운용 도표의 예

[그림 3]에서 보면, 열차 DIA 번호 #9101은 서울에서 부산을 왕복 운행하는 것으로 서울에서 부산으로 운행하는 열차번호 #3과 부산에서 서울로 운행하는 열차번호 #30으로 구성되어 있는 것을 볼 수 있다. 이때, 각 열차번호들 사이에는 열차를 재운행 하기 위해 간단한 정비시간으로 180분 이상의 대기시간이 요구되고 있다.



(그림 4) 열차 DIA 예

[그림 4]는 [그림 3]의 열차 DIA 번호 #9101을 보다 자세히 나타내고 있다. 열차 DIA를 구성하는 열차번호들에는 중량열차를 요구하는 것들이 있다. 이는 승객 수요가 많이 발생하는 시간에는 운행하는 차량 수를 늘릴 필요가 있기 때문이다. 이를 위해, [그림 4]에서 중량열차 DIA 번호 #9102가 각각 열차번호 #5와 #32로 이루어진 것을 볼 수 있다. [그림 4]와 같은 형태로 구성된 DIA도표는 철도청의 열차운용 DIA도표로 사용되고 있으나, 스케줄링의 어려움으로 운용 비용 감소를 위한 열차의 재운행을 고려하지 못한 채, 1회의 왕복 운행 스케줄 결과를 여러 날에 걸쳐서 반복하여 운행하고 있다. 본 연구에서는 이러한 문제점도 고려하여 해결하고자 하였다.

다음절에서는, 현재 철도청에서 열차운용 스

케줄링시 사용되고 있으며, 본 실험에서 사용될 제약조건들을 설명한다.

3.3 도메인 및 제약조건 분석

분석된 열차 제약조건들은 새마을호의 열차운용 DIA 작성시 사용되는 제약조건과 도메인을 기반으로 하였다.

1) 도메인(Domain) 및 변수(variable)

- 각 열차번호의 출발시간 도메인 집합을 L_i , 도착시간의 도메인 집합을 A_i 라 한다면, 운행시간 도메인 집합 DT는 다음과 같다.

$$DT = \{ i \mid (A_i - L_i), 0 \leq i \leq n \}$$

(단, i 는 정수, n 은 열차번호 총 개수)

- 각 열차역 도메인들에 대한 집합 X 는 $X = \{ \text{서울, 대구, 부산, 해운대, 울산, 포항, 마산, 진주} \}$
- 각 열차 DIA번호 도메인들에 대한 집합 TD는

$$TD = \{ TD_1, TD_2, \dots, TD_m \}$$

(단, m 은 열차 DIA 총 개수)

- 새마을호의 각 열차의 열차량 길이는, 8량과 중량열차라고 불리는 16량이 운행되고 있다. 각 열차 DIA번호 도메인 변수 TD에 대한 열차량 길이(Length())는

$$\begin{aligned} \text{Length}(TD1) &= \text{Length}(TD2) \\ &= \text{Length}(TD3) = \text{Length}(TD4) \\ &= 16 \text{ 이고,} \end{aligned}$$

$$\begin{aligned} \text{Length}(TD5) &= \text{Length}(TD6) = \\ \dots &= \text{Length}(TD27) = 8 \text{ 이다.} \end{aligned}$$

- 출발역 x_i 에서, 출발시간 L_i 에 열차가 출발해서 DT시간만큼 운행하여 A_i 시간에 도

착역 X'_i 에 도착하는 운행 관계식 R_{DT} 에 대한 열차번호들의 집합을 TB라 한다면, TB는 다음과 같다.

$$TB = \{ i \mid (X_i, L_i) R_{DT} (X'_i, A_i), 1 \leq i \leq n \}$$

여기서, i 는 총 열차 DIA 개수 n 까지의 범위이다.

2) 열차운용 문제

열차운용이라 함은 열차의 출발과 도착 시간을 나타내는 열차번호들(TB)에 여러 객차들로 구성된 하나의 가상 열차를 일컫는 열차 다이어그램(TD)을 배정하는 문제이다.

그러므로, 열차운용문제 P는 다음과 같다 (Tsang., 1993).

$$P = \langle TB, TD \rangle$$

이런 시간을 고려한 배정문제에는 여러 제약조건들이 사용된다. 다음은 철도청에서 사용하고 있는 제약조건들을 나열하겠다.

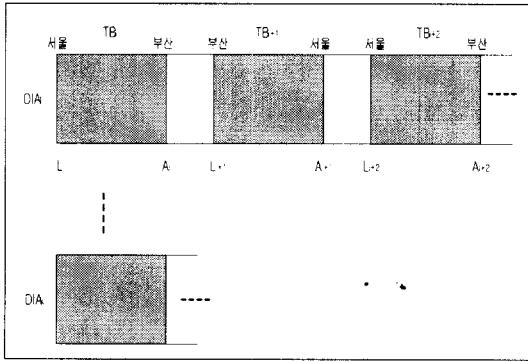
3) 제약조건

열차운용 DIA 중 하나의 열차 DIA 번호는 [그림 5]에서와 같이 2개 이상의 열차번호들을 운행한다. 임의 열차 TD가 운행하는 열차 번호들의 집합 TB를 DIA라고 하면 열차운용 DIA는 다음과 같다.

$$DIA = \{ i \mid TB_i, TB_{i+1}, \dots, i > 0 \text{인 정수} \}$$

이러한, 열차 운용 DIA들을 가지고 스케줄링 시 적용되어야할 제약조건들은 다음과 같다.

- 초기 열차의 운행 후 열차가 재운행 될 때 최소 시간 180분 이상이 필요하다. 이는 열



(그림 5) 열차 운용 DIA

차가 재운행되기 위해서는 청소시간과 식당차 음식물 운반시간, 승차시간이 필요하기 때문이다. 그러므로, 임의의 한 DIA_i 이 열차번호 TB_1 과 TB_2 를 연속하여 차례로 운행한다면, 다음과 같이 나타낼 수 있다.

$$TB_1 = (X_1, L_1) R_{DT_1}(X'_1, A_1)$$

$$TB_2 = (X_2, L_2) R_{DT_2}(X'_2, A_2)$$

이때, 먼저 운행한 TB_1 의 도착시간 A_1 과 TB_2 의 출발시간 L_2 사이의 시간 간격은 180분 이상이어야 한다. 열차번호 TB_i 의 출발시간을 $TB_i(L_i)$ 이라 하고, 도착시간을 $TB_i(A_i)$ 라고 한다면,

$$TB_i(L_i) \geq TB_{i-1}(A_{i-1}) + 180$$

(단, $TB_i \in DIA_k, 0 < k < m, 0 < i < n$)

..... [제약조건-1]

여기서 m 은 열차 DIA의 총 개수이고, n 은 열차번호의 총 개수이다.

- 반드시 모든 열차번호가 열차운용 DIA에 한 번씩 사용되어야하고, 하나의 열차운용 DIA에는 2개 이상의 열차번호 TB로 이루어져야 한다.

즉, $|DIA_k| \geq 2$ [제약조건-2]

- 하나의 열차운용 DIA에 운용된 열차번호들의 차량형식(차량길이)은 동일해야 한다. 즉, DIA_k 의 차량길이를 $Length(DIA_k)$ 라 하고, TB_i 에 요구되는 차량길이를 $Length(TB_i)$ 라 한다면, 다음과 같아야 한다.

$$Length(DIA_k) = Length(TB_i)$$

(단, $TB_i \in DIA_k, 0 < k < m, 0 < i < n$)

..... [제약조건-3]

여기서, 현재 새마을호 열차량 길이는 16량과 8량이 운행되고 있는데, 이는 시간에 따라 승객의 수요가 다르기 때문이다. 그러므로, 함수 $Length(x)$ 의 값은 8 또는 16뿐이다.

- 스케줄링 할 때, 열차운용 DIA에서 열차번호 TB들에서의 열차 출발위치와 이전의 도착위치는 동일해야 한다. 왜냐하면, 열차가 부산으로 운행하는 열차번호를 운행한 후, 서울에서 부산으로 운행하는 열차번호를 또 다시 운행할 수는 없고, 부산에서 출발하는 열차번호를 운행해야 하기 때문이다. 만약, DIA_k 가 열차번호 TB_1 과 TB_2 를 운행한다면, 다음과 같다.

$$TB_1 = (X_1, L_1) R_{DT_1}(X'_1, A_1)$$

$$TB_2 = (X_2, L_2) R_{DT_2}(X'_2, A_2)$$

그런데, 여기서 TB_1 의 도착지 X'_1 과 TB_2 의 출발지 X_2 는 동일한 역이다.

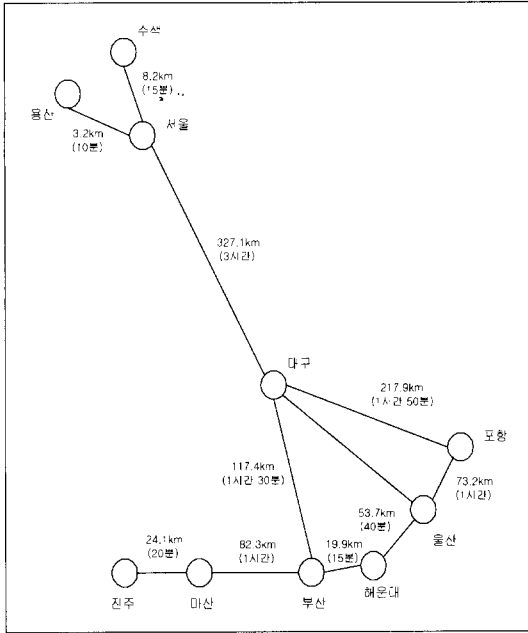
그러므로,

$$TB_j(X_j) = TB_{j+1}(X_{j+1}) \quad (\text{단, } j > 0 \text{ 정수})$$

..... [제약조건-4]

- 만약, 열차 번호를 운행하기 위한 열차가 해당 역에 없거나, 최소 대기시간 180분의 조건에 만족하는 열차가 해당 역에 없을 경우엔

인근 역에 있는 열차를 회송해야 하는 문제가 발생한다.



(그림 6) 각 구간별 위치 및 운행시간

[그림 6]은 각 구간별 회송 거리와 시간을 나타내며, 다음과 같은 조건에 만족되어질 때, 회송이 일어난다.

- IF 1) 역 X_a 에서의 열차 개수 $|X_a| = 0$
- 2) $TB_{i+1}(L_{i+1}) > TB_i(A_i) + 180$
for $TB_i \in DIA_k$

또한, 회송문제가 발생할 때에는 인근 역들에 있는 열차들 중 어느 열차를 회송해 올 것인지에 대한 문제가 발생한다. 이때는 회송 비용이 최소가 되는 것을 가져와야 한다.

즉, 역 X_a 이외의 다른 역들의 집합을 \overline{X}_a 라하고, 역 \overline{X}_a 에 있는 열차들의 집합을 $\overline{TD}_a = \{TD_1, TD_2, \dots\}$ 라 할 때, 열차 집합

\overline{TD}_a 의 중, 최소 대기시간 180분을 만족하는 열차들 중에서 대기시간이 최소인 열차를 회송열차로 선정한다.

그 회송열차들의 회송시간을 집합 $LT = \{LT_1, LT_2, \dots\}$ 라 하자. 이때, LT 의 값이 최소인 열차를 회송한다. 만약, 각 \overline{TD}_a 의 회송시간을 $\overline{TD}_a(LT_m)$ 라 한다면 회송 B에 대하여 다음과 같이 나타낼 수 있다.

$$\overline{TD}_{a_m} = \text{Minimum}(\overline{TD}_a(LT_m))$$

$$B = \langle TB_a, \overline{TD}_{a_m} \rangle$$

(단, $m > 0$ 인 정수)

..... [제약조건-5]

3.4 최종목적 함수(Goal Function)

앞에서 언급한 제약조건과 더불어 이 시스템에서 사용된 최종목적 함수를 설명한다. 본 연구에서는 다음과 같은 두 가지의 최적 목적함수를 적용하여 시스템에 적용하였다.

첫 번째, 전체 운용 비용 값이 최소가 되는 열차 운용 DIA 목록을 최적의 해로 생각한다.

역 X_a 에서의 열차 대기시간을 WT_a , 열차 회송시 회송열차의 회송시간을 LT_a 라 하면, 비용 방정식(Cost Function)은 다음과 같다.

$$C_a = 4 \cdot LT_a + WT_a$$

여기서 사용된 대기시간과 회송시간 비율은 철도 스케줄 전문가에 의해 아래처럼 제시된 비율이다.

$$LT : WT = 4 : 1$$

이 비용 값이 모든 열차의 배정 P문제에 적용된다면 다음과 같다.

$$\sum_{ij}^m (4 \cdot LT_{ij} + WT_{ij})$$

(단, i, j는 서로 다른 열차, m은 열차운용 DIA 총 개수)

이 비용 값들 중, 최소값이 되는 열차운용 DIA의 목록을 구하는 문제는

$$\text{Minimize } \sum_{ij}^m (4 \cdot LT_{ij} + WT_{ij})$$

..... [목적함수-1]

이다.

마지막으로, 열차운용 DIA의 개수를 최소화함으로써, 주어진 열차로 가능한 많은 열차번호를 운용하는 것을 목적으로 하는 문제이다.

$$\text{Minimize } (|DIA|)$$

..... [목적함수-2]

4. 실험 및 결과분석

실험에서 사용하고 비교한 기본 열차 정보는 다음과 같다.

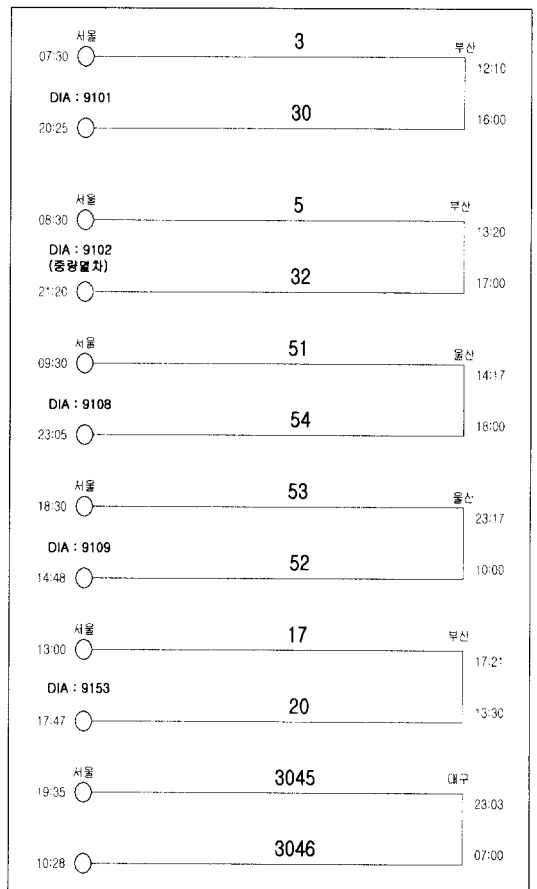
- 열차편성에 따른 열차번호
:중량 편성(16량)된 열차는 각 운행노선에 대하여 다음과 같이 운행한다.
 - 하행선 : #5, #9, #1, #7
 - 상행선 : #32, #36, #26, #34
- 각 차량 사무소에서 관리하는 열차 운용 DIA 수는 다음과 같다.
 - 서울 : 19 DIA
 - 부산 : 8 DIA

또한, 열차 시간표(열차번호) 데이터는 1998년 5월에 운행되었던 경부선 새마을호의 운행 시간표를 사용하였고, 이러한 동일 데이터를 가지고

3일간의 스케줄링 결과를 인간 전문가의 스케줄링 결과와 비교하였다.

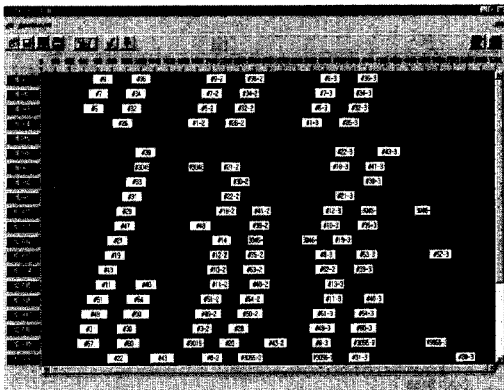
먼저, 앞에서 언급한 동일한 데이터를 가지고 인간 전문가에 의해 스케줄링된 결과는 [그림 7]의 형태로 그려져서 사용되고 있으며, 이는 일부분만을 나타낸 것이다.

실제로 철도청에선, 이와 같이 1일간 스케줄링된 열차운용 DIA를 여러 날에 걸쳐서 반복 사용하고 있다. 더욱이, 스케줄링의 어려움과 복잡함 때문에 열차 운용에 있어서 가장 비용을 최소화시키는 최적함수를 최대로 고려하기란 여



(그림 7) 인간 전문가에 의해 작성된 열차운용 DIA표의 예

간 힘든 것이 아니며, 실제로 거의 고려하지 않고 스케줄링 되고 있다. 특히, 이러한 철도청의 시간 데이터로 3일간 운행된다면 27개의 열차 DIA 전부가 운용되어야하며, 하나의 열차 DIA는 최대 6개의 열차번호를 운행하게 된다. 또한, 스케줄링된 결과를 알아보기도 힘이 든다. 예를 들어, [그림 7]을 보면, 열차번호 #9101은 서울에서 오전 8시에 출발해서 부산에 12시 10분에 도착한 후, 다시, 16시에 출발하여 20시 10분에 서울로 돌아오는 하루동안의 운행 열차라는 것을 쉽게 알 수 있다. 그러나, 열차 DIA 번호 #9153의 경우에는, 실제 2일간에 걸쳐서 운행하는 것을 알 수 있다. 이와 같은 알아보기 힘든 복잡한 구조로 형성된 시간표와 더불어, 각 역에 그날그날 도착한 열차들을 고려하여 재 스케줄링 하기란 여간 힘든 것이 아니다. 실제로, 이러한 스케줄링을 하기 위해서는 여러 명의 스케줄 전문가들이 수일동안 작업을 해야했다.

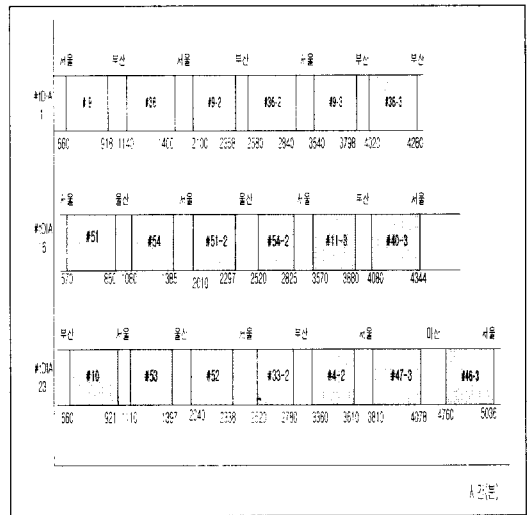


[그림 8] 간트차트의 예

본 논문에서 구현한 시스템은 ILOG사의 ILOG Scheduler, ILOG Solver, ILOG View를 개발도구로 사용하였고, 사용된 시스템 환경은 Pentium 133MHz, RAM 128Mbyte의 시스템이다.

구현된 시스템으로 앞에서 언급한 데이터를 실험한 결과, 수행시간은 시스템 시간 16.5초가 걸렸다. 실험결과는 [그림 8]과 같다. 이는 각 시간대 별로 출력된 간트차트(Gantt Chart)로 일부만을 나타낸 것이다. 이와 같이, 스케줄 결과를 간트차트인 GUI로 제공함으로써 사용자가 쉽게 알아볼 수 있도록 하였다.

길게 나열된 간트차트의 출력된 결과물 중 일부분을 열차 운용 DIA 도표로 나타낸다면 [그림 9]와 같다.



[그림 9] 열차운용 스케줄 결과로 형성된 열차 운용 DIA도표

[그림 9]에서, 중량 열차(16량)인 열차 DIA 번호 #1은 서울과 부산을 왕복 운행하며, #9, #36, #9-2, #36-2, #9-3, #36-3인 6개의 중량 열차를 요구하는 열차번호들로 구성되어졌음을 알 수 있으며, 열차 운용 DIA 번호 #23번은 부산, 서울, 울산, 마산을 운행하는 7개의 열차번호들로 구성되어진 것을 볼 수 있다. 게다가 일부 열차 DIA의 경우 최대 9개까지 운용되었고, 열차 DIA 1개

(표 1) 과거 철도청 스케줄과 본 시스템에 의한 실험 결과 비교

항 목	과거 철도청 스케줄 결과	본 시스템에 의한 실험 결과
스케줄 최적화 함수	고려하지 않음	최소 운용 비용 목적함수와 최소 열차 DIA 운용 목적함수 적용
도착한 열차의 재 스케줄 적용	재 스케줄 어려움	재 스케줄 고려
스케줄 기간 (대기시간 고려여부)	1일간의 스케줄 결과를 반복 사용(열차의 매일 운행 종료와 시작간 대기시간 고려 안함)	전체 스케줄 기간을 처리 → 본 실험은 3일간의 데이터를 사용(열차의 매일 운행 종료와 시작간 대기시간을 고려함)
운영된 열차 DIA 수	27개 전부 사용	26개 사용(1개의 열차 DIA 미사용)
스케줄 결과 표시	수작업에 의한 DIA 도표	GUI 제공(간트차트)
스케줄링 시간	다수의 전문가에 의해 수일에서 수주간 스케줄링	시스템 시간 16.5초(3일분)

(열차 DIA번호 #5)가 전혀 사용되지 않았다. 이 경우, 현재의 데이터에서는 1량의 열차를 운영할 필요가 없거나, 필요한 열차 시간표를 추가 편성하여 열차 DIA를 추가로 운영할 수 있다는 뜻이다. 이는, 과거 철도청 스케줄은 1일간의 스케줄 결과를 반복하여 사용하고 있기 때문에 열차의 1일 운행 종료 후와 다음날 운행 시작 사이의 열차 대기 시간을 스케줄링시 고려하지 못하였으나, 본 스케줄링 시스템에서는 열차 운용을 높이고자 이러한 대기 시간과 도착된 열차에 대해서도 재 스케줄링을 고려하였기 때문에 위와 같은 결과가 나타났다. 또한, 열차운용 비용 측면에 있어서는 비용을 최소화시키는 최적함수가 고려됨으로써 열차운용 비용이 가장 적게 스케줄링 되었고, 열차 운용 면에 있어서는 열차운용 DIA 개수를 최소화시키는 최적함수가 고려됨으로써 1개의 열차 DIA를 운용하지 않아도 모든 열차 번호들을 운행하는 스케줄 결과를 얻을 수 있었다. [표 1]은 과거 철도청 스케줄의

결과와 본 시스템을 사용한 실험결과를 비교한 내용이다.

5. 결론 및 향후 연구과제

철도청 스케줄링 전문가들이 열차 운용 스케줄링을 하기 위해서는 다수의 전문가들이 많은 시간과 노력을 감수해야만 하며, 전문가의 실수로 인해 오류가 발생할 가능성이 있다. 뿐만 아니라, 복잡한 스케줄링의 특성상 경제적으로 비용이 적게 드는 스케줄 결과를 이끌어 내기가 어렵다. 스케줄 결과를 나타내는데 있어서는 복잡함으로 인해 열차운용 DIA표를 구성하기가 어렵다. 그러나, 본 연구에서는 컴퓨터의 자동 수행으로 오류 발생 가능성이 거의 없으며, 최적 함수를 적용함으로 가장 적은 열차 DIA 개수로 주어진 열차번호들을 모두 운용하였으며, 비용이 최소가 되도록 운용 스케줄링 하였다.

또한, CSP 해결기법을 적용한 언어를 사용함으로써 해를 구하는 시간이 수초 밖에 걸리지 않았고, 간트차트라는 적절한 GUI를 제공함으로써 사용자가 쉽게 열차운용 스케줄링 결과를 인식하도록 하였다.

이와 같은 열차운용 스케줄링 시스템은, 스케줄링 향상은 물론, 전문가들의 실수에 의해 발생하는 오류를 줄이며, 요구되는 열차 자원을 줄임으로써 운용비용 감소, 수익 증가와 같은 경제적인 파급효과를 가져온다. 또한 스케줄링 생성에 대한 표준적인 방법을 제시하여 새로운 스케줄링 실무자가 쉽게 스케줄링 시뮬레이션을 할 수 있는 AI 기반 스케줄링 시스템을 제공할 수도 있다. 이는 열차 승무원 스케줄링 시스템, 차량 배정 스케줄링 시스템, 열차신호제어시스템과 같은 다른 시스템으로의 파급효과도 크다. 더불어, 본 연구는 열차운용에서의 선진운영기법 도입 및 현 업무와 한국 실정에 맞는 법제도 등을 고려한 전문가시스템 개발에 의의가 있다.

향후, 본 연구에서는 사용자에 의해 스케줄이 제어될 수 있으며, 시스템이 동적인 환경에 유연히 대처 할 수 있는 대화형 시스템(interactive system)으로의 전환을 모색하고자 하며, 본 연구에 고려 안된 부분인 열차 운용 스케줄링에 있어서, 가상의 열차 DIA에 실제 차량을 할당하는 차량 스케줄을 병행할 필요가 있다. 또한 열차운용 스케줄링 전문가로부터 열차운용 스케줄링 시에 사용된 제약조건들과 법제도 등을 지속적으로 이끌어 냄으로서 보완할 필요가 있다.

감사의 글

이 연구는 1997년 인하대학교 교내 연구비 지원에 의해 이루어졌으며 이에 감사드립니다.

참고 문헌

- 양종윤, “스케줄링 문제 해결을 위한 지식 기반 기법과 제약 만족 기법의 비교 연구”, *인하대 석사 논문*, 1998.2.
- Dhar. Vasant, Nicky Ranganathan, “Integer Programming vs. Expert Systems : An Experimental Comparison”, *ACM, Vol.33, No.3*, 323~336, 1990.
- Ernesto M.Morgado, João P.Martins, “CREWS_NS: Scheduling Train Crew in The Netherlands”, *AAAI*, 1997, 893~902.
- Frost D., R.Dechter, “In search of the best constraint satisfaction search”, *AAAI Press*, 1994.
- Jo Geun-Sik, Jong-Jin Jung, Chang-Yoon Yang, “Expert System for Scheduling in an Airline Gate Allocation”, *Expert Systems with Applications, Vol.13, No.4*, An International Journal, Elsevier Science Ltd., NOV, 275~282, 1997.
- Lamma. E, Mello. P & Milano. M, “A distributed constraint-based scheduler”, *Artificial Intelligence in Engineering 11*, 91~105, 1997.
- Mackworth A.k., “Consistency in Networks of Relations”, *AAAI Press*, 1997.
- Murphy. Kathleen, Elizabeth Ralston, David Friedlander-Brightware, Inc. Rodney Swab, Paul Steege-Union Pacific Railroad, “The Scheduling of Rail at Union Pacific Railroad”, *AAAI Press*, 903~912, 1997.
- Nadel B.A., “Constraint Satisfaction Algorithm”, *Computational Intelligence 5*, 1989.
- Pascal Van Hentenryck, “Contraint Satisfaction in Logic Programming”, *MIT Press*, U.S.A, 1989.
- Schrijver. Alexander, “Minimum Circulation of Rail way Stock”, *CWI Quarterly, Vol. 6, No. 3*, 205~

217, 1993.9.

Tsang, Edward, *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.

Wallace R.J., "Why AC-3 is Almost always Better than AC-4 Establishing Arc Consistency in CSPs", *AAAI Press*, 1994